CONFIG MANAGEMENT

MIT VAULT UND SPRING CLOUD

adidas

#DBC | Digital IT, 29. März 17


DANIEL EICHTEN
DOMAIN ARCHITECTURE
CODES (MOSTLY) IN JAVA
ENJOYS GOOD COFFEE
LOVES FUJIFILM
@DANIELEICHTEN
@DEICHTEN
adidas

PRODUCTS MADE FROM OCEAN PLASTICS

ROBOTS MAKING SHOES

PITCH

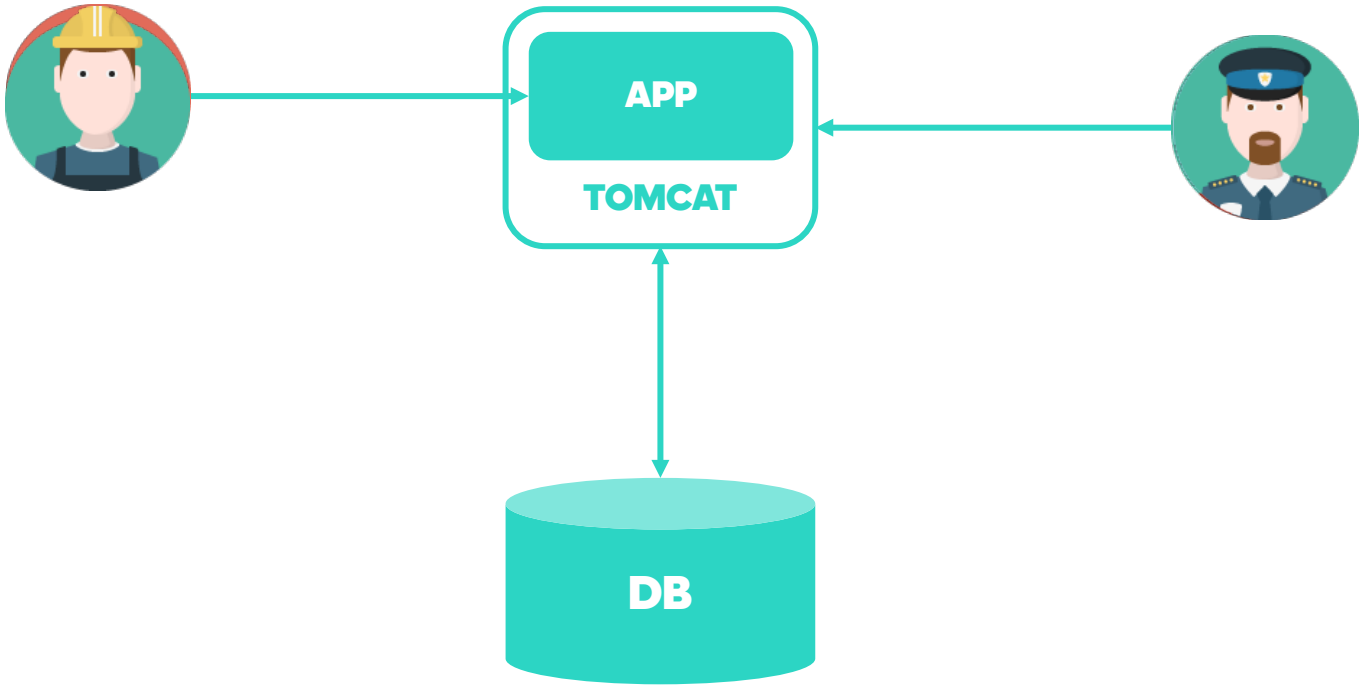

MAKERLAB

WHITEBOARD ROOM

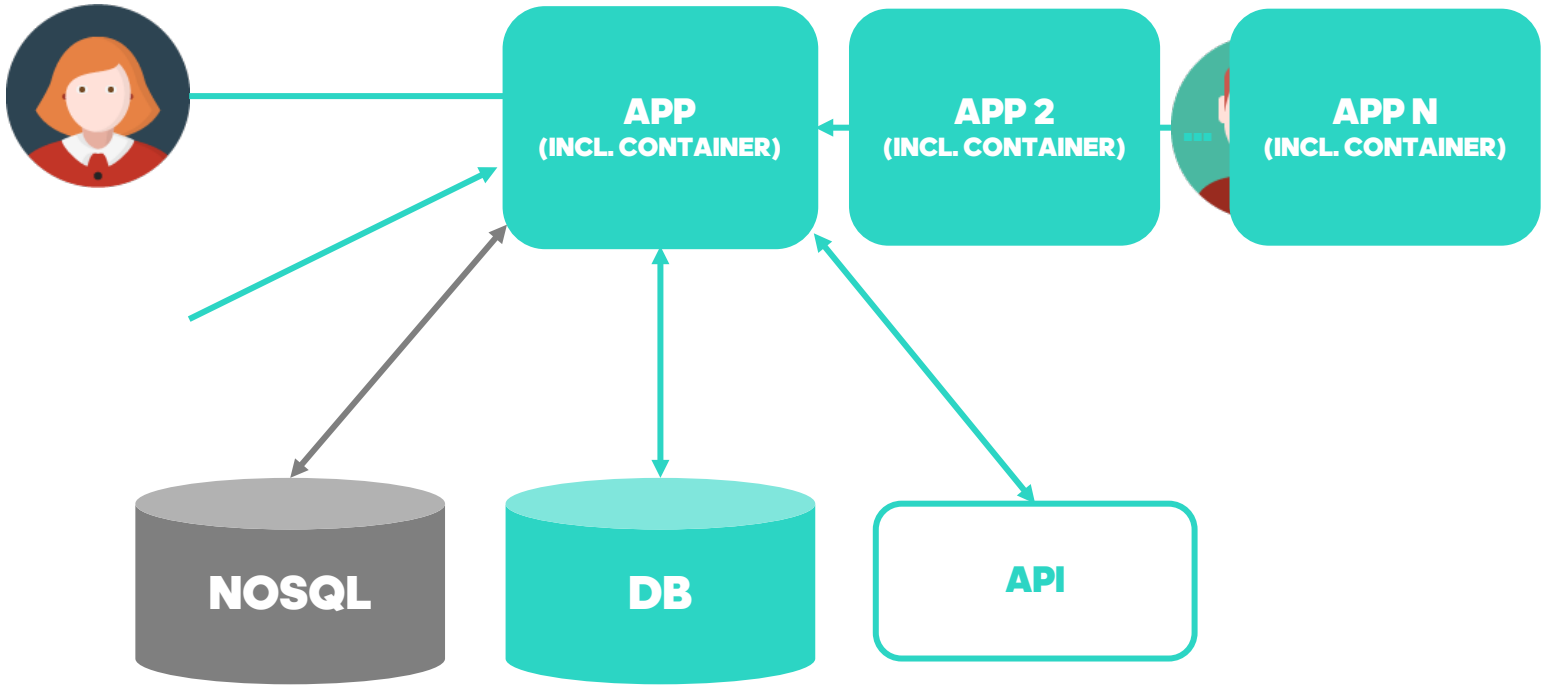
GUESTS PERFORMING A HAKA

1 INTRO

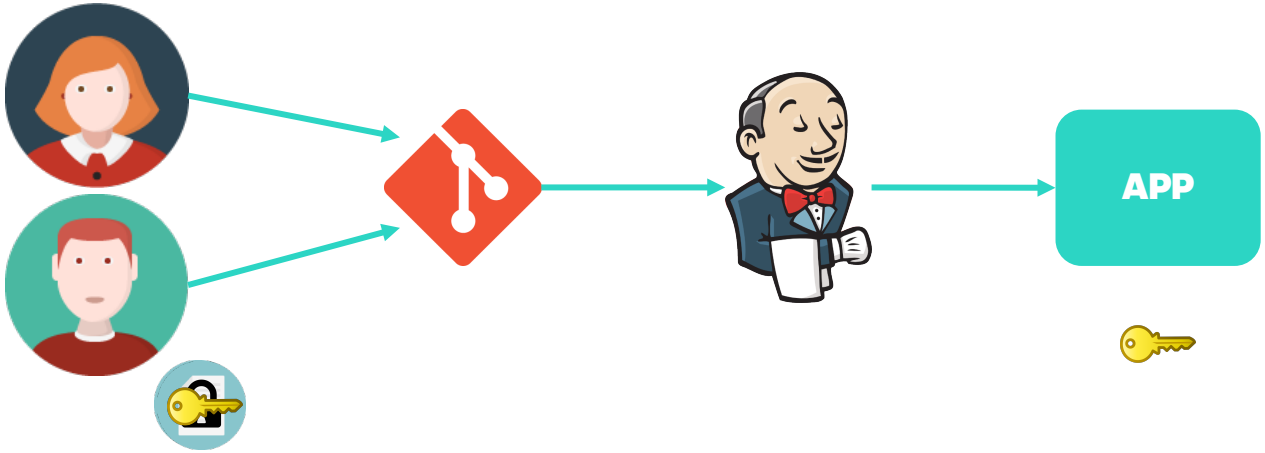
SECRETS MANAGEMENT PROBLEM STATEMENT



SECRETS MANAGEMENT PROBLEM STATEMENT



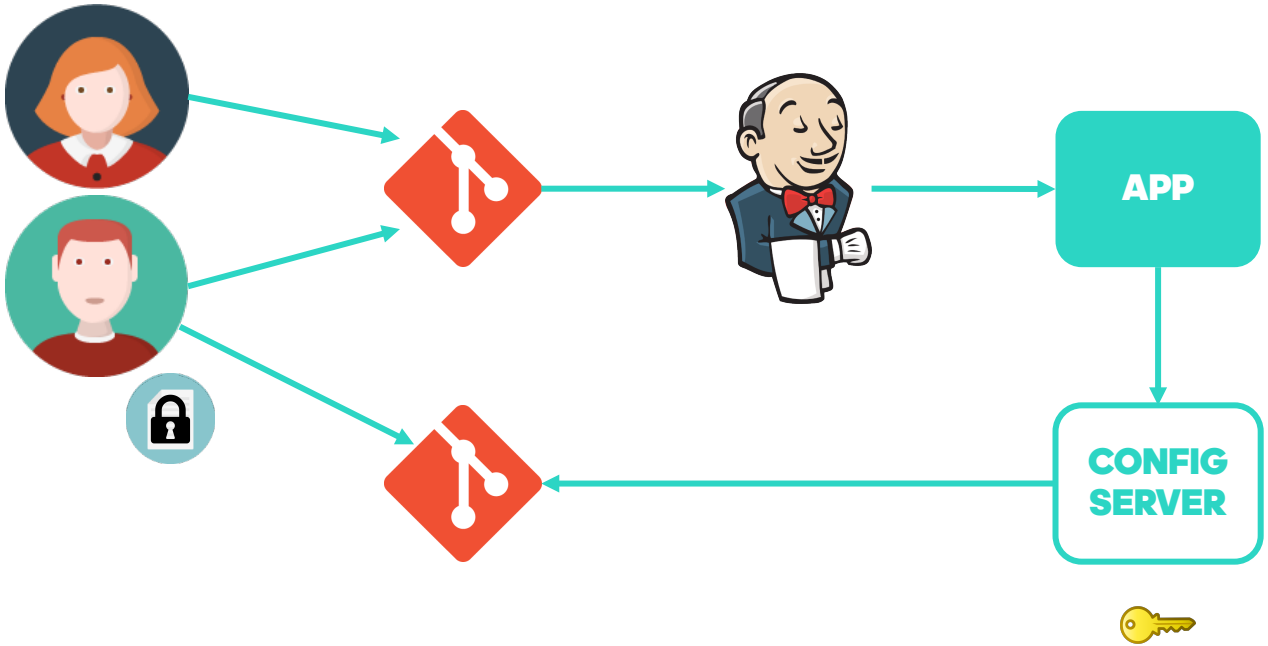
SECRETS MANAGEMENT PROBLEM STATEMENT



CHICKEN-EGG-PROBLEM

2 SPRING CLOUD CONFIG

SPRING CLOUD CONFIG




STOP
DEMO TIME

SPRING CLOUD CONFIG DEMO

SPRING CLOUD CONFIG VORTEILE

ZENTRALE ABLAGE

- Git als default Option
- SVN und Vault ebenso möglich
- Kombination verschiedener Repositories möglich

VERSCHLÜSSELUNG

- Symmetrische und asymmetrische Verschlüsselung
- Management von mehreren Schlüssels gleichzeitig

MANAGEMENT

- Abruf in verschiedenen Formaten
- Durch einfaches Interface Unterstützung außerhalb der Spring-Welt
- Push Notification bei Config Updates

ABER...

...IMMER NOCH NICHT PERFEKT!

3 | VAULT

WAS IST VAULT?

A TOOL FOR MANAGING SECRETS

VAULT FEATURES

SECRET STORAGE

- Storage or dynamic creation of secrets
- All data is encrypted
- All dynamically created secrets do have a lease time with automatic revocation afterwards
- Strict access conrtol via ACL

KEY ROLLING

- On the fly encryption of updated secrets stored in vault
- Key Rolling of internally used keys
- Enforce key rolling of dynamically created secrets via lease lifetime

FULL AUDIT TRAIL

- Every client interaction (authentication, token creation, secret access, secret revocation, ...) regardless of success or failure is logged
- Build in log shipping to various backends
- Combining lease feature with short lifetimes will help to identify leaks

... EINIGES MEHR!

VAULT STRUKTUR

- APPROLE
- AWS EC2
- GITHUB
- LDAP
- MULTI-FACTOR
- OKTA
- RADIUS
- CERTIFICATES
- TOKEN
- USER / PASS

AUTH

CORE

AUDIT

- FILE
- SYSLOG
- SOCKET

SECRETS

- AWS
- CASSANDRA
- CONSUL
- CUBBYHOLE
- GENERIC
- MONGO
- MSSQL
- MYSQL
- CERTIFICATES
- POSTGRES
- RABBITMQ
- SSH
- TRANSIT
- YOUR OWN


STOP
DEMO TIME

VAULT DEMO

4

SPRING && VAULT

SPRING && VAULT

SPRING-VAULT

- Low-Level und High-Level Abstraktion
- Authentifizierung mittels verschiedener Mechanismen
- Schreiben, Lesen, Löschen & Object-Mapping
- Version 1.0.0 RC1 seit 16.03.2017

SPRING-CLOUD-VAULT-CONFIG

- Client-side Support zum Lesen von Secrets aus Vault
- Sub-Module zum dynmischen Erzeugen von Secrets zu AWS, Cassandra, Consul, MongoDB, MySQL, PostgreSQL und RabbitMQ

SPRING VAULT – VAULTTEMPLATE

```
@Configuration
class VaultConfiguration extends AbstractVaultConfiguration {

   @Override
   public VaultEndpoint vaultEndpoint() {
      return new VaultEndpoint();
   }

   @Override
   public ClientAuthentication clientAuthentication() {
      return new TokenAuthentication("…");
   }
}
```

SPRING VAULT – VAULTTEMPLATE

```
public class Example {

   // inject the actual template
   @Autowired
   private VaultOperations operations;

   public void writeSecrets(String userId, String password) {
      Map<String, String> data = new HashMap<String, String>();
      data.put("password", password);
      operations.write(userId, data);
   }

   public Person readSecrets(String userId) {
      VaultResponseSupport<Person> response = operations.read(userId, Person.class);
      return response.getBody();
   }
}
```

SPRING VAULT – PROPERTYSOURCE

```
@VaultPropertySource(value = "aws/creds/s3",
   propertyNamePrefix = "aws.“,
   renewal = Renewal.RENEW)
public class MyConfig { }

public class Example {

   // inject the actual values
   @Value("${aws.access_key}“)
   private String awsAccessKey;

   @Value("${aws.secret_key}“)
   private String awsSecretKeyKey;

   public InputStream getFileFromS3(String filenname) {
   // …
   }
}
```

SPRING CLOUD VAULT CONFIG - BOOTSTRAPPING

```
<dependency>
   <groupId>org.springframework.cloud</groupId>
   <artifactId>spring-cloud-starter-vault-config</artifactId>
</dependency>

<repositories>
   <repository>
      <id>spring-milestones</id>
      <name>Spring Milestones</name>
      <url>https://repo.spring.io/libs-milestone</url>
      <snapshots>
         <enabled>false</enabled>
      </snapshots>
   </repository>
</repositories>
```


STOP
DEMO TIME

SPRING VAULT CONFIG DEMO

5 | EXTRAS

KEYBASE.IO INIT


Q
&
A

THANK YOU

adidas

#Digital Brand Commerce

CONTACTS

#DBC

Daniel Eichten

#DBC | Digital IT
e-mail: daniel.eichten@adidas-group.com
twitter: @danieleichten
keybase: deichten
github: deichten